

CNS Assignment No.3

```
#include<iostream>
#include<string.h>
using namespace std;
class crc
{
    string input,divisor,divident,result;
    int len_divident,len_gen,len_inp;
public:
    string fun_xor(string a, string b)
    {
        string result="";
        if(a[0]=='0')
            return a.substr(1);
        else;
        {
            for(int i=0;i<len_gen;i++)
            {
                result=result+(a[i]==b[i]?'0':'1');
            }
            return result.substr(1);
        }
    }

    void modulo_div()
    {
        string temp_div=divisor;
        string temp_divident=divident.substr(0,len_gen);
        int j=len_gen;
        while(j<len_divident)
        {
            temp_divident=fun_xor(temp_divident,temp_div);
            temp_divident=temp_divident+divident[j];
            j++;
        }
        result=input+fun_xor(temp_divident,temp_div);
    }

    void getdata()
    {
        cout<<"\nEnter Input: ";
        cin>>input;
        cout<<"\nEnter coefficient of genarator polynomial: ";
        cin>>divisor;
        len_gen=divisor.length();
        len_inp=input.length();
        divident=input;
        int r=len_gen-1;
        for(int i=0;i<r;i++)
        {
            divident=divident+'0';
        }
    }
}
```

```

        len_divident=divident.length();
        modulo_div();
    }

    void sender_side()
    {
        cout<<"\nInput: "<<input;
        cout<<"\nPolynomial: "<<divisor;
        cout<<"\nDivident: "<<divident;
        cout<<"\nData to send: "<<result;
    }

    void receiver_side()
    {
        string data_rec;
        cout<<"\nEnter Data Received: ";
        cin>>data_rec;
        string temp_div=divisor;
        string temp_divident=data_rec.substr(0,len_gen);
        int j=len_gen;
        while(j<data_rec.length())
        {
            temp_divident=fun_xor(temp_divident,temp_div);
            temp_divident=temp_divident+data_rec[j];
            j++;
        }
        string error=fun_xor(temp_divident,temp_div);
        cout<<"reminder is: "<<error;
        bool flag=0;
        for(int i=0;i<len_gen-1;i++)
        {
            if(error[i]=='1')
            {
                flag=1;
                break;
            }
        }
        if(flag==0)
            cout<<"\nCorrect Data Received Without Any Error";
        else
            cout<<"\nData Received Contain Some Error";
    }
};

int main()
{
    crc c;
    c.getdata();
    c.sender_side();
    c.receiver_side();
    return 0;
}

```

C:\Users\rashm\Downloads\CNS Ass3 program.exe

Enter Input: 10101001

Enter coefficient of genarator polynomial: 1011

Input: 10101001

Polynomial: 1011

Divident: 10101001000

Data to send: 10101001000

Enter Data Received: 10101001000

reminder is: 000

Correct Data Received Without Any Error

Process exited after 50.45 seconds with return value 0

Press any key to continue . . .

CNS Assignment No.4

```
# include <iostream>
# include <conio.h>
# include <stdlib.h>
# include <time.h>
# include <math.h>
# define TOT_FRAMES 500
# define FRAMES_SEND 10
using namespace std;
class gobkn
{
    private:
        int fr_send_at_instance;
        int arr[TOT_FRAMES];
        int arr1[FRAMES_SEND];
        int sw;
        int rw;           // tells expected frame
    public:
        gobkn();
        void input();
        void sender(int);
        void reciever(int);
};

gobkn :: gobkn()
{
    sw = 0;
    rw = 0;
}

void gobkn :: input()
{
    int n; // no of bits for the frame
    int m; // no of frames from n bits
    cout << "Enter the no of bits for the sequence no ";
    cin >> n;
    m = pow (2 , n);
    int t = 0;
    fr_send_at_instance = (m / 2);
    for (int i = 0 ; i < TOT_FRAMES ; i++)
    {
        arr[i] = t;
        t = (t + 1) % m;
    }
    sender(m);
}

void gobkn :: sender(int m)
{
    int j = 0;
    for (int i = sw ; i < sw + fr_send_at_instance ; i++)
    {
```

```

        arr1[j] = arr[i];
        j++;
    }

    for (int i = 0 ; i < j ; i++)
        cout << " SENDER : Frame " << arr1[i] << " is sent\n";
    reciever (m);
}

void gobkn :: reciever(int m)
{
    time_t t;
    int f;
    int f1;
    int a1;
    char ch;
    srand((unsigned) time(&t));
    f = rand() % 10;
    // if = 5 frame is discarded for some reason
    // else they are correctly recieved
    if (f != 5)
    {
        for (int i = 0 ; i < fr_send_at_instance ; i++)
        {
            if (rw == arr1[i])
            {
                cout << "RECIEVER : Frame " << arr1[i] << " recieved correctly\n";
                rw = (rw + 1) % m;
            }
            else
                cout << "RECIEVER : Duplicate frame " << arr1[i] << " discarded\n";
        }
        a1 = rand() % 15;
        // if a1 belongs to 0 to 3 then
        // all ack after this (incl this one) lost
        // else
        // all recieved
        if (a1 >= 0 && a1 <= 3)
        {
            cout << "(Acknowledgement " << arr1[a1] << " & all after this lost)\n";
            sw = arr1[a1];
        }
        else
            sw = (sw + fr_send_at_instance) % m;
    }

    else
    {
        f1 = rand() % fr_send_at_instance;
        // f1 gives index of the frame being lost
        for (int i = 0 ; i < f1 ; i++)
        {
            if (rw == arr1[i])
            {
                cout << " RECIEVER : Frame " << arr1[i] << " recieved correctly\n";
            }
        }
    }
}

```

```

        rw = (rw + 1) % m;
    }
    else
        cout << " RECIEVER : Duplicate frame " << arr1[i] << " discarded\n";
}

int ld = rand() % 2;
// ld == 0 frame damaged
// else frame lost
if (ld == 0)
    cout << " RECIEVER : Frame " << arr1[f1] << " damaged\n";
    cout << " (Frame " << arr1[f1] << " lost)\n";
    for (int i = f1 + 1 ; i < fr_send_at_instance ; i++)
        cout << " RECIEVER : Frame " << arr1[i] << " discarded\n";
    cout << " (SENDER TIMEOUTS --> RESEND THE FRAME)\n";
    sw = arr1[f1];
}
cout << "Want to continue...";
cin >> ch;
if (ch == 'y')
    sender(m);

else
    exit(0);
}

int main(int, int)
{
    gobkn gb;
    gb.input();
    getch();
    return 0;
}

```

C:\Users\rashm\Downloads\CNS Ass4 program.exe

Enter the no of bits for the sequence no

5

SENDER : Frame 0 is sent

SENDER : Frame 1 is sent

SENDER : Frame 2 is sent

SENDER : Frame 3 is sent

SENDER : Frame 4 is sent

SENDER : Frame 5 is sent

SENDER : Frame 6 is sent

SENDER : Frame 7 is sent

SENDER : Frame 8 is sent

SENDER : Frame 9 is sent

SENDER : Frame 10 is sent

SENDER : Frame 11 is sent

SENDER : Frame 12 is sent

SENDER : Frame 13 is sent

SENDER : Frame 14 is sent

SENDER : Frame 15 is sent

SENDER : Frame 16 is sent

SENDER : Frame 17 is sent

SENDER : Frame 18 is sent

SENDER : Frame 19 is sent

SENDER : Frame 20 is sent

SENDER : Frame 21 is sent

SENDER : Frame 22 is sent

SENDER : Frame 23 is sent

SENDER : Frame 24 is sent

SENDER : Frame 25 is sent

RECIEVER : Duplicate frame 0 discarded

RECIEVER : Duplicate frame 1 discarded

RECIEVER : Duplicate frame 2 discarded

RECIEVER : Duplicate frame 3 discarded

RECIEVER : Duplicate frame 4 discarded

RECIEVER : Duplicate frame 5 discarded

RECIEVER : Duplicate frame 6 discarded

RECIEVER : Duplicate frame 7 discarded

RECIEVER : Duplicate frame 8 discarded

RECIEVER : Duplicate frame 9 discarded

RECIEVER : Duplicate frame 10 discarded

RECIEVER : Frame 11 recieved correctly

RECIEVER : Frame 12 recieved correctly

RECIEVER : Frame 13 recieved correctly

RECIEVER : Frame 14 recieved correctly

RECIEVER : Frame 15 recieved correctly

Want to continue...

n

Process exited after 28.67 seconds with return value 0

Press any key to continue . . .

CNS Assignment No.5

```
import java.io.*;
import java.net.InetAddress;
public class Subnet1
{
    public static void main(String[] args) throws IOException
    {
        System.out.println("ENTER IP:");
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        String ip = br.readLine();
        String checkclass = ip.substring(0, 3);
        int cc = Integer.parseInt(checkclass);
        String mask = null;
        if (cc > 0)
        {
            if (cc <= 127)
            {
                mask = "255.0.0.0";

                System.out.println("Class A IP Address");
                System.out.println("SUBNET MASK:\n" + mask);
            }

            else if (cc >= 128 && cc <= 191)
            {
                mask = "255.255.0.0";

                System.out.println("Class B IP Address Used for Multicasting");
                System.out.println("SUBNET MASK:\n" + mask);
            }

            else if (cc >= 192 && cc <= 223)
            {
                mask = "255.255.255.0";

                System.out.println("Class C IP Address Used for Multicasting");
                System.out.println("SUBNET MASK:\n" + mask);
            }

            else if (cc >= 224 && cc <= 239)
            {
                mask = "255.0.0.0";

                System.out.println("Class D IP Address Used for Multicasting");
            }

            else if (cc >= 240 && cc <= 254)
            {
                mask = "255.0.0.0";

                System.out.println("Class E IP Address Experimental Use");
            }
        }
    }
}
```



```

    }

    String networkAddr = "";
    String lastAddr = "";
    String[] ipAddrParts = ip.split("\\.");
    String[] maskParts = mask.split("\\.");
    for (int i = 0; i < 4; i++)
    {
        int x = Integer.parseInt(ipAddrParts[i]);
        int y = Integer.parseInt(maskParts[i]);
        int z = x & y;
        networkAddr += z + ".";
        int w = z | (y ^ 255);
        lastAddr += w + ".";
    }

    System.out.println("First IP of Block:" + networkAddr);

    System.out.println("Last IP of Block:" + lastAddr);
}
}

```

C:\Windows\System32\cmd.exe

Microsoft Windows [Version 10.0.19045.3208]
(c) Microsoft Corporation. All rights reserved.

C:\Users\rashm\Desktop>javac Subnet1.java

C:\Users\rashm\Desktop>java Subnet1

ENTER IP:

192.168.0.1

Class C IP Address Used for Multicasting

SUBNET MASK:

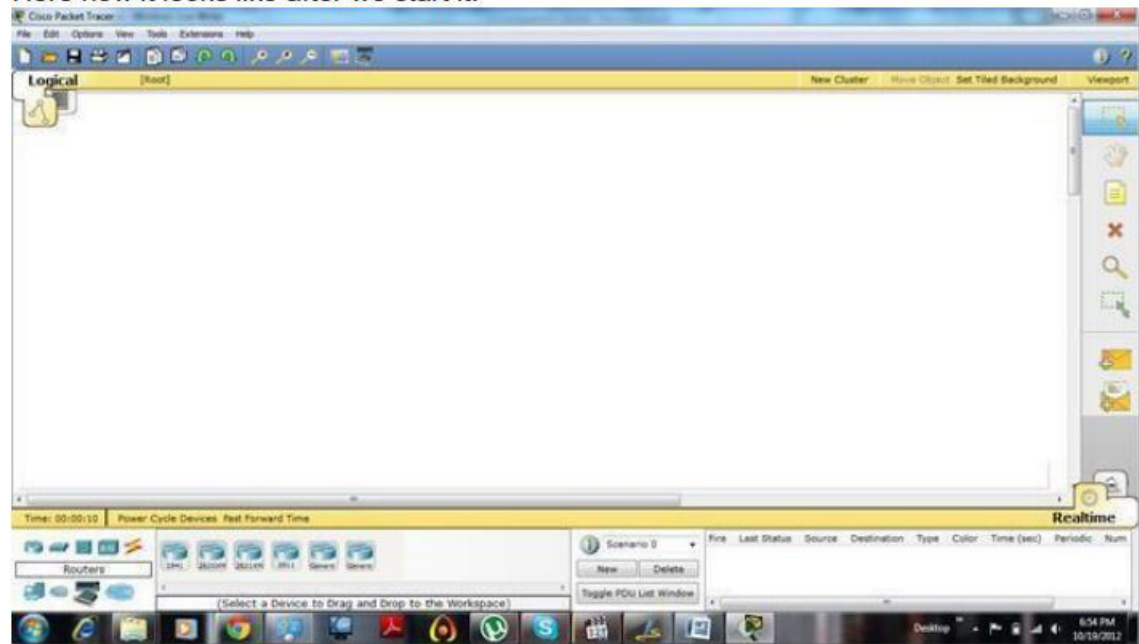
255.255.255.0

First IP of Block:192.168.0.0.

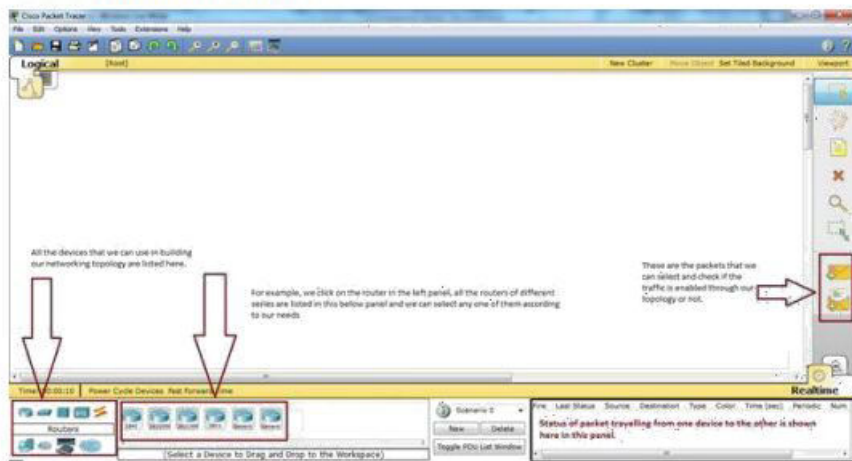
Last IP of Block:192.168.0.255.

C:\Users\rashm\Desktop>_

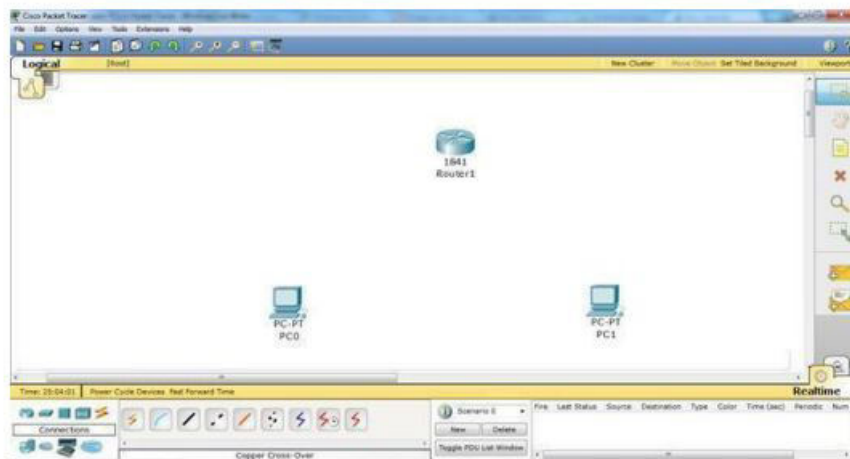
Here how it looks like after we start it.



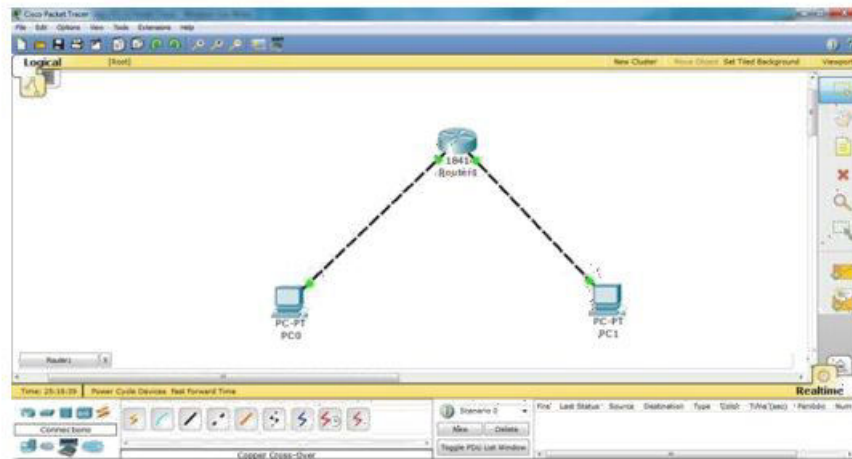
We are different modules and panels available in the packet tracer. Some important modules, which are important to understand for the working in Packet Tracer, are mentioned in the following diagram.



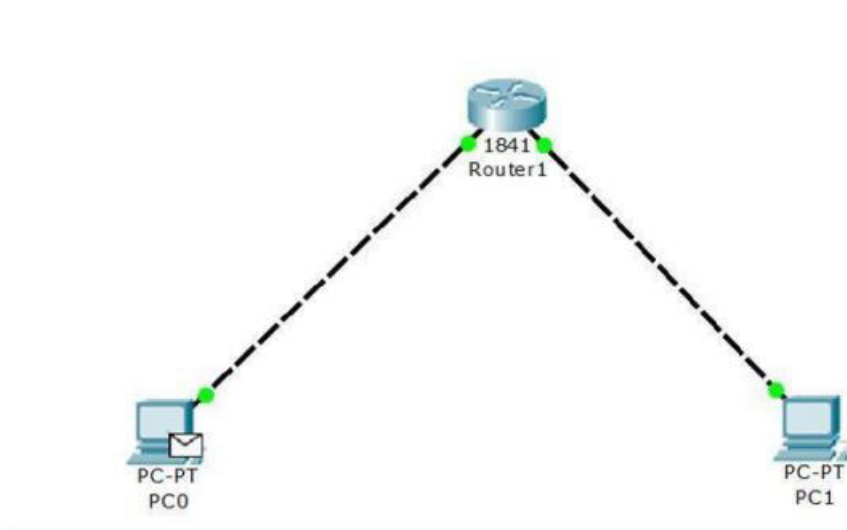
Now, in order to create a topology, we will have to select some of the devices and put them in our main window i.e. the white portion of packet tracer. and here how it looks after we add the devices.



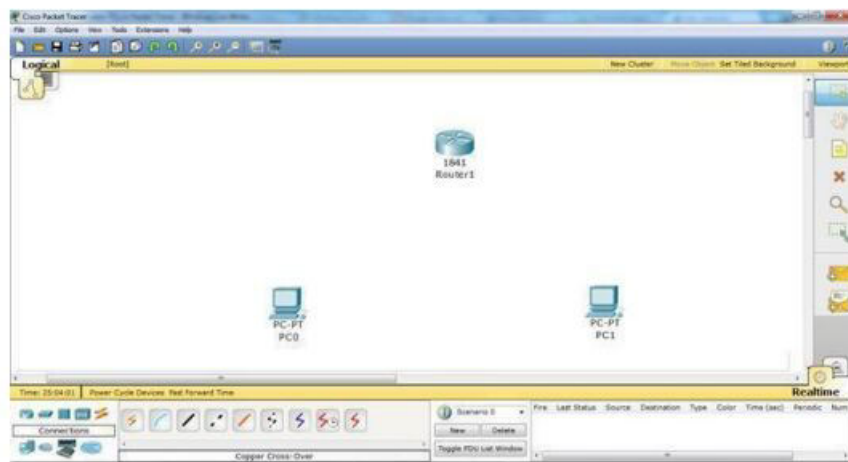
Now, we will have to connect these devices and for that we use cables. To understand the cables, please refer to my following [article](#).



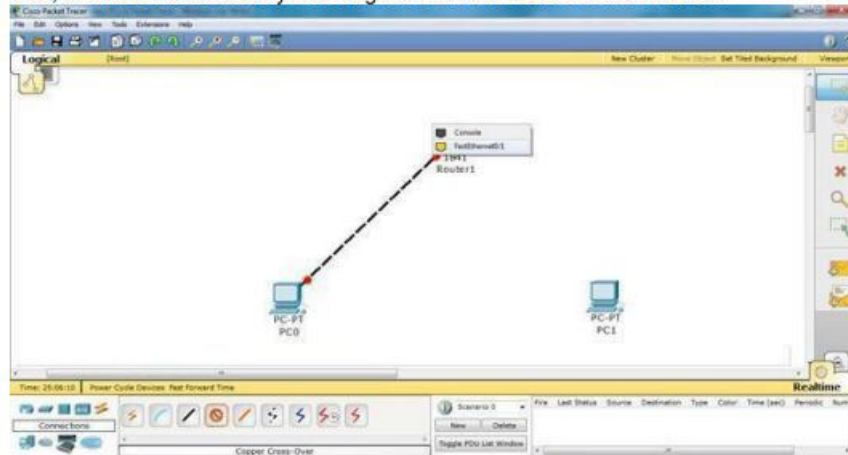
And after you successfully create the topology, you can check either the traffic is flowing or not by selecting the packet from right panel and putting it on both PCs as follows.



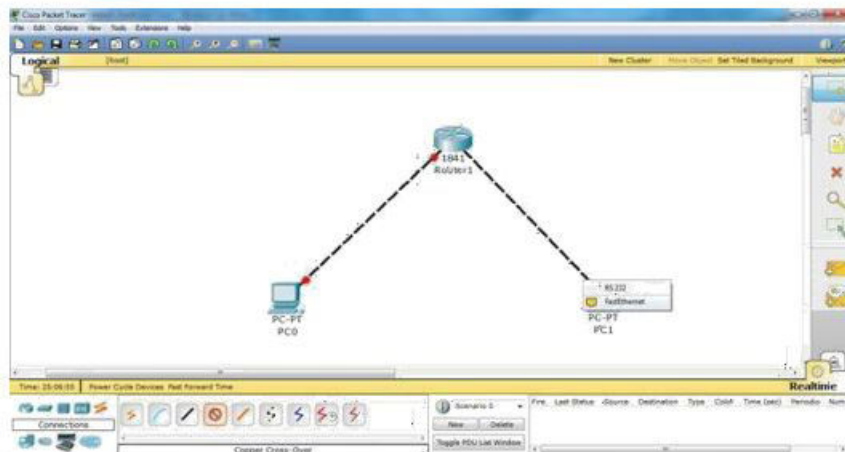
For detailed article on making the topology and successfully enabling the communication, Here, we will see communication enabled between PCs via Router in Packet Tracer. So, for this we need two PCs, a router, and two cross over cables to connect them. Important point is that we use cross over cable to connect PC to a router because they both use the same pins for transmission and receiving of data.



Now, we will connect them by selecting fast ethernet interfaces on both ends.



Similarly, on the PC side we will select fast Ethernet interface.



Now, we have connect the devices. Further, we will go to the router CLI mode and enter the following commands. Step by step, we will have to do the following things.

- i. Access the interfaces one by one
- ii. Assign IP addresses to interfaces
- iii. Change the status of the interfaces i.e. from Down to Up.
- iv. Assign IP addresses to PCs.
- v. Assign Default GateWay to PCs. FYI fast ethernetip address is the gateway address to the PC. Now, commands of the Router CLI mode are as follows.

```

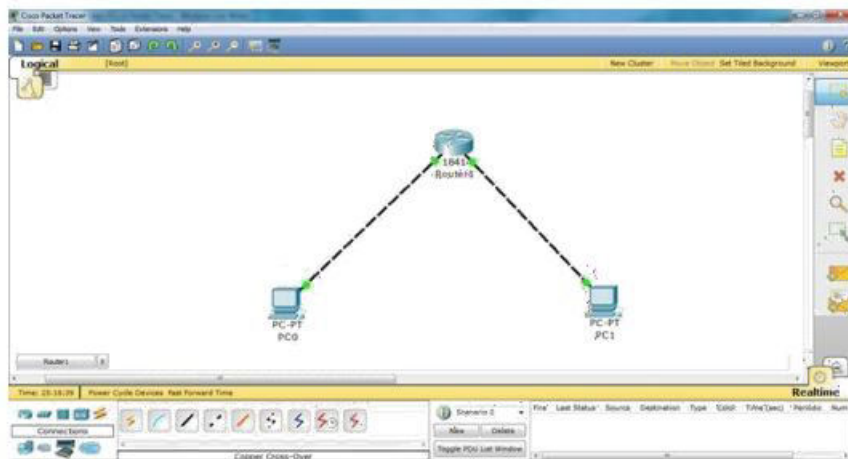
R1>en
Password:
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#inte
R1(config)#interface fa
R1(config)#interface fastEthernet 0/0
R1(config-if)#ip ad
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#no shutdown

%LINK-3-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state t
o up

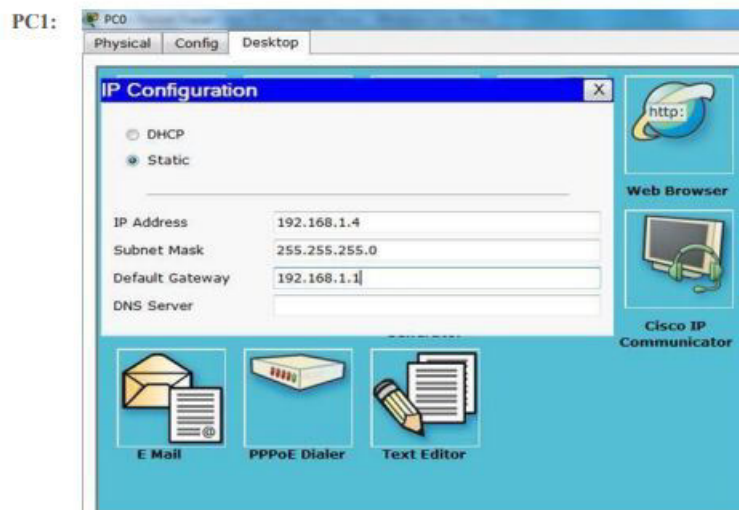
R1(config-if)#exit
R1(config)#interfa
R1(config)#interface fastEthernet 0/1
R1(config-if)#ip address 192.168.2.1 255.255.255.0
R1(config-if)#no shutdown

%LINK-3-CHANGED: Interface FastEthernet0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state t
o up
  
```

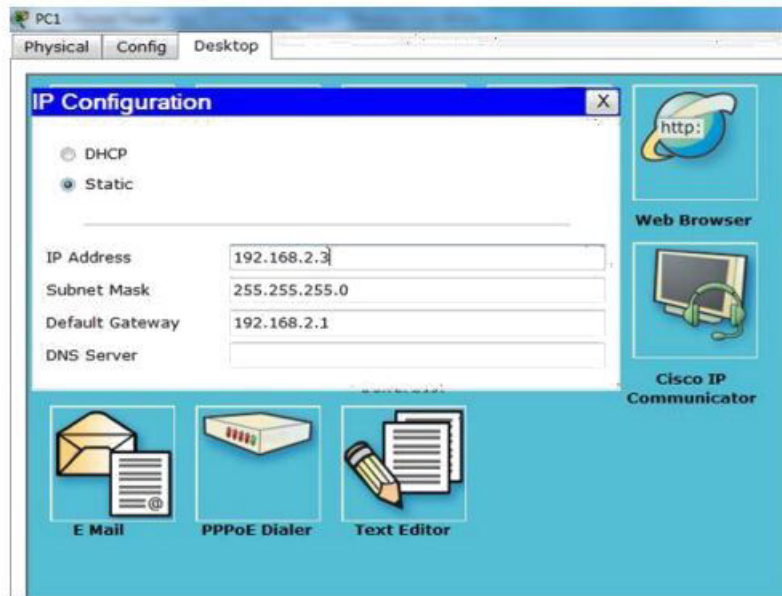
Now, we have accessed both interfaces one by one and we have assigned IP addresses respectively.



See the difference the lights have changed the color from Red to Green :) Now, let's assign IP addresses to the PCs. Click on PC1, go to Desktop, then click IP Configuration.

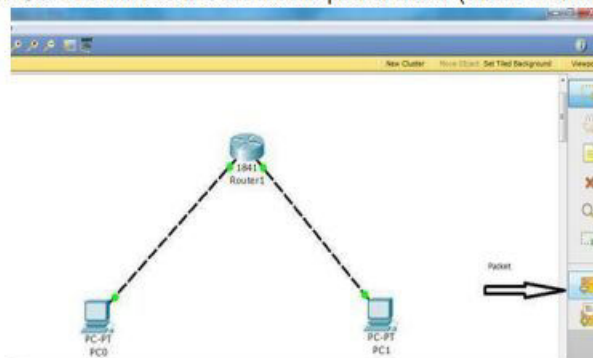


PC2:



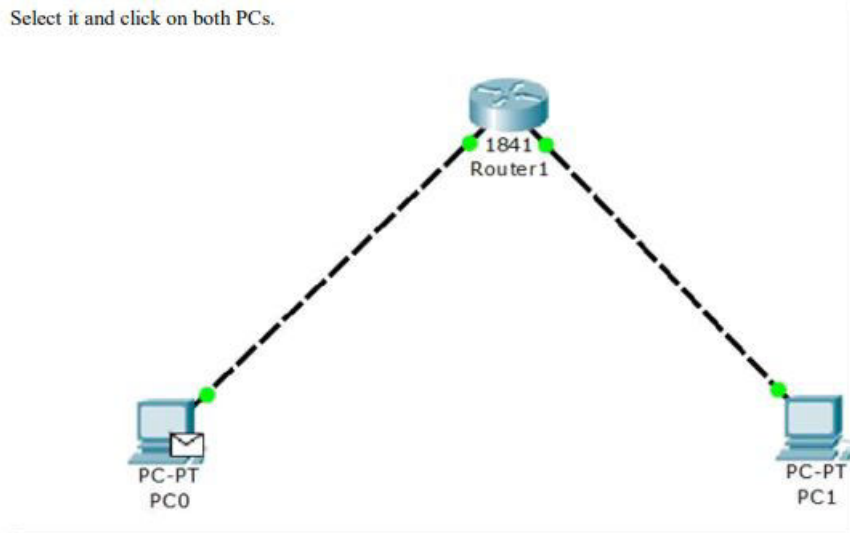
Now, our communication is enabled and we are able to communicate from PC1 to PC2 via Router.

Click on the packet in the right panel on the packet tracer, then click on PC1 and then click on PC2. You will see the successful packet tracer (status is shown in the bottom



right corner)

Select it and click on both PCs.



CNS Assignment No.7(Calculator)

Client(Python)

```
#Client.py
import socket
import sys
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_address = ('localhost', 10000)
sock.connect(server_address)
print("Connecting to %s port %s" % server_address)
while True:
    message = input("Enter the equation: ")
    print("Calculating result\n")
    sock.sendall(message.encode())
    data = sock.recv(16)
    print("Result is: %s" % data.decode())
    a = input("\nDo you wish to continue?(Yes/No): ")
    if a == 'No':
        break
print("Closing connection")
sock.close()
```

CNS Assignment No.7(Calculator)

Server(Python)

#Server.py

```
import socket
import sys
import select

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_address = ('localhost', 10000)
sock.bind(server_address)
print("Starting up on %s port %s" % server_address)
sock.listen(1)

while True:
    print("Waiting for connection")
    connection, client_address = sock.accept()
    print("Connection from ", client_address)
    ready = select.select([connection], [], [], 15)
    if ready[0]:
        try:
            while True:
                equation = connection.recv(16)
                print("Received: ", equation)
                if equation:
                    print("Sending the result back to the client")
                    result = eval(str(equation.decode()))
                    connection.sendall(str(result).encode())
                else:
                    print("No more operations from the client")
                    break
            finally:
                connection.close()
```

C:\Windows\System32\cmd.exe

Microsoft Windows [Version 10.0.19045.3208]
(c) Microsoft Corporation. All rights reserved.

C:\Users\rashm\Desktop>python Client.py

Connecting to localhost port 10000

Enter the equation: 2+2

Calculating result

Result is: 4

Do you wish to continue?(Yes/No): Yes

Enter the equation: 65+65

Calculating result

Result is: 130

Do you wish to continue?(Yes/No): Yes

Enter the equation: 120-51

Calculating result

Result is: 69

Do you wish to continue?(Yes/No): Yes

Enter the equation: 45*2

Calculating result

Result is: 90

Do you wish to continue?(Yes/No): Yes

Enter the equation: 150/2

Calculating result

Result is: 75.0

Do you wish to continue?(Yes/No): No

Closing connection

C:\Users\rashm\Desktop>

C:\Windows\System32\cmd.exe - python Server.py

Microsoft Windows [Version 10.0.19045.3208]
(c) Microsoft Corporation. All rights reserved.

C:\Users\rashm\Desktop>python Server.py

Starting up on localhost port 10000

Waiting for connection

Connection from ('127.0.0.1', 61440)

Received: b'2+2'

Sending the result back to the client

Received: b'65+65'

Sending the result back to the client

Received: b'120-51'

Sending the result back to the client

Received: b'45*2'

Sending the result back to the client

Received: b'150/2'

Sending the result back to the client

Received: b''

No more operations from the client

Waiting for connection

CNS Assignment No.8(Chat)

Client(java)

```
import java.net.*;
import java.io.*;
import java.util.Scanner;
public class Client
{
    public static void main (String args[])
    {
        Scanner sc=new Scanner(System.in);
        String send, receive;

        try
        {
            DatagramSocket dsoc=new DatagramSocket(9099);
            byte sbuf[]=new byte[1024];
            byte rbuf[]=new byte[1024];

            while(true)
            {
                System.out.print("Client: ");
                send=sc.nextLine();
                sbuf=send.getBytes();
                DatagramPacket spkg = new DatagramPacket(sbuf,sbuf.length,InetAddress.getByName("127.0.0.1"),10000);
                dsoc.send(spkg);
                DatagramPacket rpkg = new DatagramPacket(rbuf,rbuf.length);
                dsoc.receive(rpkg);
                System.out.print("\nServer: "+new String(rpkg.getData())+"\n");

                for(int i=0;i<1024;i++)
                {
                    rbuf[i] = '\0';
                }
            }

            catch(Exception e)
            {
                System.out.println(e);
            }
        }
    }
}
```

CNS Assignment No.8(Chat)

Server(java)

```
import java.net.*;
import java.io.*;
import java.util.Scanner;
public class Server
{
    public static void main (String args[])
    {
        Scanner sc=new Scanner(System.in);
        String send, receive;

        try
        {
            DatagramSocket dsoc=new DatagramSocket(10000);
            byte sbuf[]=new byte[1024];
            byte rbuf[]=new byte[1024];

            while(true)
            {
                DatagramPacket rpkg = new DatagramPacket(rbuf, rbuf.length);
                dsoc.receive(rpkg);
                System.out.print("Client: "+new String(rpkg.getData())+"\n");

                for(int i=0; i<1024; i++)
                {
                    rbuf[i] = '\0';
                }
                System.out.print("\nServer: ");
                send=sc.nextLine();
                sbuf=send.getBytes();
                DatagramPacket spkg = new DatagramPacket(sbuf, sbuf.length, InetAddress.getByName("127.0.0.1"), 9099);
                dsoc.send(spkg);
            }

            catch(Exception e)
            {
                System.out.println(e);
            }
        }
    }
}
```



```
C:\Windows\System32\cmd.exe - java Server
Microsoft Windows [Version 10.0.19045.3208]
(c) Microsoft Corporation. All rights reserved.
```

```
C:\Users\rashm\Desktop>javac Server.java
```

```
C:\Users\rashm\Desktop>java Server
Client: Hello There!!!
```

```
Server: Hey!!, Hii...
Client: I'M SONiC, and You??
```

```
Server: I'M LUCiFER :)
Client: Ohhhh Nicee..
```

```
Server: Yeahh ;)
```

C:\Windows\System32\cmd.exe - java Client

Microsoft Windows [Version 10.0.19045.3208]

(c) Microsoft Corporation. All rights reserved.

C:\Users\rashm\Desktop>javac Client.java

C:\Users\rashm\Desktop>java Client

Client: Hello There!!!

Server: Hey!!, Hii...

Client: I'M SONiC, and You??

Server: I'M LUCiFER :)

Client: Ohhhh Nicee..

Server: Yeahh ;)

Client:

CNS Assignment No.9

```
import java.net.*;
import java.util.*;
public class IPDemo
{
    public static void main(String[] args)
    {
        String host;
        Scanner ch=new Scanner(System.in);
        System.out.print("1.Enter Host Name \n2.Enter IP Address \nChoice= ");
        int choice=ch.nextInt();
        if(choice==1)
        {
            Scanner input=new Scanner(System.in);
            System.out.print("\nEnter Host Name: ");
            host=input.nextLine();
            try
            {
                InetAddress address=InetAddress.getByName(host);
                System.out.println("IP Address-> "+ address.getHostAddress());
                System.out.println("Host Name-> "+ address.getHostName());
                System.out.println("Host Name And IP Address-> "+ address.toString());
            }
            catch(UnknownHostException ex)
            {
                System.out.println("Could not find "+ host);
            }
        }
        else
        {
            Scanner input = new Scanner(System.in);
            System.out.print("\nEnter IP Address: ");
            host = input.nextLine();
            try
            {
                InetAddress address=InetAddress.getByName(host);
                System.out.println("Host Name-> "+ address.getHostName());
                System.out.println("IP Address-> "+ address.getHostAddress());
                System.out.println("Host Name And IP Address-> "+ address.toString());
            }
            catch(UnknownHostException ex)
            {
                System.out.println("Could not find "+ host);
            }
        }
    }
}
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.3208]
(c) Microsoft Corporation. All rights reserved.

C:\Users\rashm\Desktop>javac IPDemo.java

C:\Users\rashm\Desktop>java IPDemo
1.Enter Host Name
2.Enter IP Address
Choice= 1

Enter Host Name: www.google.com
IP Address-> 142.250.183.196
Host Name-> www.google.com
Host Name And IP Address-> www.google.com/142.250.183.196

C:\Users\rashm\Desktop>java IPDemo
1.Enter Host Name
2.Enter IP Address
Choice= 2

Enter IP Address: 8.8.8.8
Host Name-> dns.google
IP Address-> 8.8.8.8
Host Name And IP Address-> dns.google/8.8.8.8

C:\Users\rashm\Desktop>
```